JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 129 - 137
https://publishoa.com
ISSN: 1309-3452

Unifying Architecture To Generate Descriptions: Features To Text

Ajay Kumar Yadav
Department of Computer Science
New Horizon College of Engineering
Bangalore,India
yadavajay2055@gmail.com

Aniket Kumar Yadav
Department of Computer Science
New Horizon College of Engineering
Bangalore,India
Aniketyadav98650@gmail.com

Dipak Yadav
Department of Computer Science
New Horizon College of Engineering
Bangalore,India
dipak.yadav5501@gmail.com

Ms. Soja Naveen
(Sr.Assistant Professor)
Department of Computer Science
New Horizon College of Engineering
Bangalore,India
sojars@newhorizonindia.edu

Dr. Pamela Vinitha
(Assistant Professor)
Department of Computer Science
New Horizon College of Engineering
Bangalore,India
Pamela.vinitha@gmail.com

ABSTRACT

Automatically interpreting visuals is one of the challenges that has plagued Artificial Intelligence (AI). It connects the two domains of computer vision and natural language understanding. We employ recent advances in neural networks, such as CNN and RNN, to deliver the finest captions in this research. The model that is single end to end to predict the caption given a photo which unifies the two architecture to create the text utilizing the features. Two forms of discriminator architectures (CNN and LSTM-based structures) are introduced, each with its unique set of benefits. The variety of inscriptions created was forced to a breaking point by these approaches. There should be no assumptions about explicit preconditions in the model. Instead of relying on predetermined forms, standards, or classes, you must figure out how to construct sentences from the preparatory data. The accuracy of the model is proved by comparing it to numerous datasets. Many evaluation indications show that our model is highly accurate. Our model is validated using the benchmark datasets Flickr8K and Flickr30K. One of the approaches used to evaluate is BLEU scores.

Keywords— Deep Learning, Neural Networks, Architecture, Text Description.

I. INTRODUCTION

Artificial Intelligence (AI) is presently at the centre of the innovation economy, and this project is built on the same foundation. In recent years, a branch of AI known as Deep Learning has gained a lot of attention due to its excellent accuracy outcomes when compared to previous Machine Learning algorithms. The challenge of generating a meaningful language from an image is tough, yet it can have a significant impact, such as assisting the visually impaired in better understanding images.

We can now develop models that could generates captions for images thanks to advances in deep learning techniques, the availability of large datasets, and computer power.. It can have a great impact for visually impaired people. It can also be useful in the field of automating the job of person interpreting the image. It will be widely used in the field where text is mostly used where you can infer or generate text from the image. It can be also helpful in the video analysis frame by frame. Social Media Platforms can also infer from the images directly.

Deep learning which is a branch of machine learning which uses layers of artificial neural networks to mimic human neural networks in automatically decoding intuition from large amounts of data. It differs from other machine learning algorithms that rely heavily on feature engineering and use domain knowledge to create feature extractors. Simple features in the early levels are reconstructed from one layer to another in order to produce complex characteristics in the stack layers of neural networks. As a result, modelling and training deeper networks is computationally intensive, prompting the development of more complex computer chips like as Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs). Recurrent neural networks (RNNs), autoencoders, convolutional neural networks (CNNs), deep belief networks (DBNs), and deep Boltzmann machines are some of the most prominent deep learning models currently available (DBM). Because of its shift and space invariant qualities, the convolutional neural network is the most ideal deep learning algorithm for interpreting visual data, taking use of hierarchical learning in integrating basic patterns to generate complex patterns and structures.

Each filter represents different properties of the input data using the shared weights architectural pattern of filters, which when added together can generate more complicated structures.

Picture captioning is a much more difficult task than image classification, which has been the emphasis of the computer vision field. The relationship between the objects in an image must be captured in an image description. In addition to visual comprehension of the image, the above semantic knowledge must be conveyed in a natural language such as English, necessitating the use of a language model. Previous attempts have always included stitching the two models together. This task is quite hard than the previously famous image classification. Here we have to express the object relativity to each other as well the activities that are involved in. So, we have to understand the semantic knowledge as well to express in natural language, which entails combining visual comprehension with a linguistic model.

Previous initiatives have looked into merging existing solutions to the aforementioned problems in order to create visual descriptions. We'd like to demonstrate our single joint model, which uses a picture to predict the intended word sequence. The task's main inspiration originates from the machine's most recent developments in translation. The goal is to use the Recurrent Neural Network (RNN) to translate more simply while keeping the model's cutting-edge performance. The "Encoder" i.e RNN read the original sentence before converting to a vector representation, which is the hidden states for the "Decoder" i.e RNN, which generate the desired sentences.

II. RELATED WORK

Image caption generation is the important aspect of scene comprehension since it uses can be in a number of application (e.g, image searching, narrating story from album, assisting visually impaired individuals with online navigation, and so on). Many differences picture captioning approached that has been develop throughout the year.

The architecture i.e. employed by ILSVRC winner had made significant contributions to this discipline. The VGG16 proposed by He. et al. was one such architecture that we used. Aside from that, machine translation research has continually aided in increasing the state of the art metrics in sentence production.

Image captioning was done using a pipeline approach by Microsoft's AI Lab. They created high-level characteristics for each putative object in the image using a CNN. They then utilised Multiple Instances of Learning (MIL) to know about the region that best corresponded to each words present. On MSCOCO, the results were 21.7 percent BLEU scores. Following this pipeline technique, Google researcher develops the primary end-to-end trainable model. The RNN model utilized in machine translation served as inspiration.

Because CNN features are commonly employed altogether computer vision tasks, Vinyals et al. that replaced encoder i.e RNN in image CNN feature. This model is understood as Neural Image Caption (NIC). Referencing this two Stanford researcher alters the NIC. The learning were about the inter-modal correspondences between linguistic and visual data employing a technique that relies on image databases and sentence descriptions. Their multimodal embedding alignment technique used a unique combinations of Convolutional Neural Network rather than picture area, bidirectional Recurrent Neural Networks in place of phrases, and a structure that aim to correlate the 2 modalities. They achieved state-ofthe-art results using the Flickr8K, Flickr30K, and MSCOCO datasets. Jonathan et al. tweaked their model even more. Once they offered a depth captioning job in each of image region were recognised and a group of description was produced. Wang et al. introduced next model which uses a convolutional neural networks (CNN) and 2 independent LSTM network. Xu et al. presented one of the most recent works that was inspired by the NIC model. Inspired by advancement in machine translation & visual detection, they developed a attentionbased models which automatically learned to describe the content in image.Not only have image captioning models advanced in recent years, but so have many evaluation measures. The BLEU score was the accuracy statistic we used. CIDEr, developed by Vedantam et al., is gradually replacing BLEU, which were a common assessment for metric embraced by most of the group.

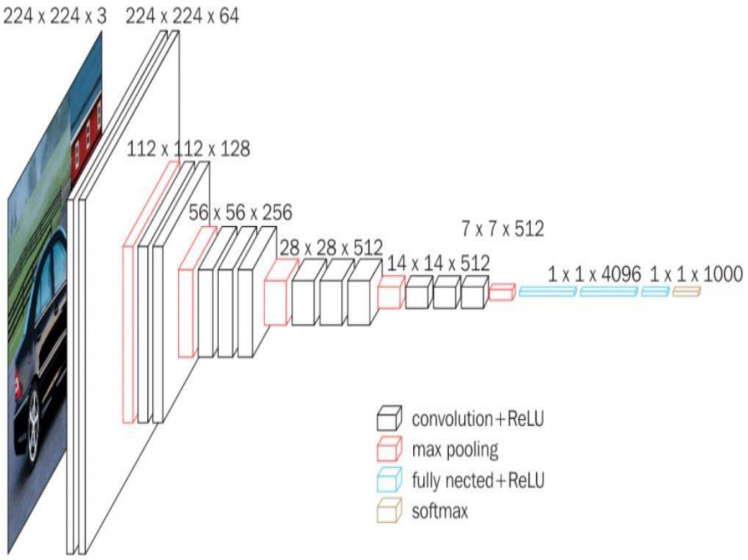
III. MODEL ARCHITECTURE

A. DEEP CNN ARCHITECTURE:

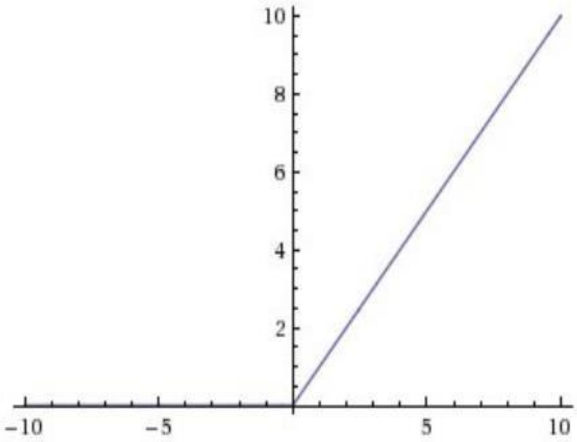
Convolutional Neural Network (ConvNet or CNNs) are a type of Artificial Neural Network which has shown to be particularly good at picture recognition and classification. Object identification, selfdriving cars, image captioning, and other tasks have all made substantial use of them. It is a type of neural network that has accepted input in 2D shape. Because images may be presented in a similar way, it makes sense to use neural networks in picture pre-processing. Convolution is distinct which generates some feature that can be detected in a image. It filter inputs before producing feature map which summarises the detection of feature in the input. These network learn filter in context of the given prediction problem during training. The output of the filter using the first input array.

There is only one value for time. The features are a twodimensional vector that is produced when the filter is applied. Once those are built, they are sent to the feature map via nonlinearity, such as ReLU, the outputs of the completely connected layers.

The image taken for processing must be 224*224 images. The preprocessing i.e done by subtracting the mean RGB value from each pixel determined by training the image.



The convolution layer are made up of 3*3 filter with a stride length of 1. A 2*2-pixel window with a stride lengths of 2 is used for maximum pooling. All of the photos must be transformed to a 224*224pixel image. Each convolution layer has a Rectified Linear Unit (ReLU) activations functions. The function f(x) = max(0, x) is computed by ReLU. The following is the ReLU function's output:



A ReLU layer has the advantage of speeding up the stotachastic gradient descent over sigmoid and tanh layers. Unlike more sophisticated operations (exponential, etc.), the ReLU operation is straight forward to implement by setting an activations matrix to zero. Our needs, however, don’t have to categorise the images, so we remove last layer of 1*1*1000 classification layer.

As a result, output of CNN encoder produces 1*1*4096 encode signal, which is sent to RNN that creates the language. More successful CNN frameworks, such as Resnet, have been developed, however they are computationally highly expensive due to Resnet’s 152 layers compared to vgg16’s,16.

The entire training approach for the Convolution Network can be summarised below:

Step 1- All filter, parameters, and weight are given random values .

Step 2- This network accept a training images as an input and calculates the output probabilities for each class using forward propagations (convolutional, ReLU, & pooling operation in the Connected layer, as well in forward propagation in FullyConnected layers.

Let's consider the output probability for the boats images are [0.1, 0.2, 0.4, 0.1]. The output probabilities is also random because the weights in the initial training example are allocated at random.

Step 3-In step three, calculate total errors at the output layers in step three (summation of tha all 4 classes).

Total Errors = $\sum$ ½ (target probability – output probability) ^{2}

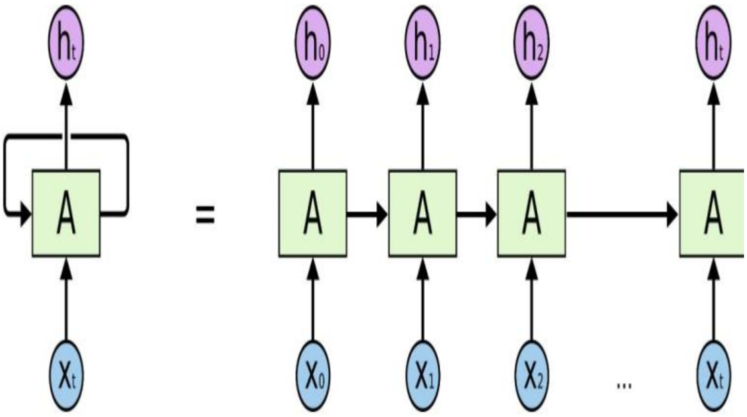
Step 4- Using back propagation, calculate error gradient for all weight in the networks, then apply gradient descent for updating all the filter value / weight and parameter value to minimise output errors. The weight are changed in according to how muchof a contribution they make to the overall inaccuracy.

When the two image is fed again, the output probability may be [0.2, 0.2, 0.5, 0.3], which is nearer to the targeted vectors [0, 1, 0, 0].

This indicates that the networks had learned to correctly categorise that particular images by modifying its weight / filter to lower output error.
Number of filter& filter size, network architectures, and on was all fixed before Step 1 & did not vary during this training process only in the values of the filters matrix and connection weight change.
Step 5- Steps 2,3,4 should be repeated for each image in the training.

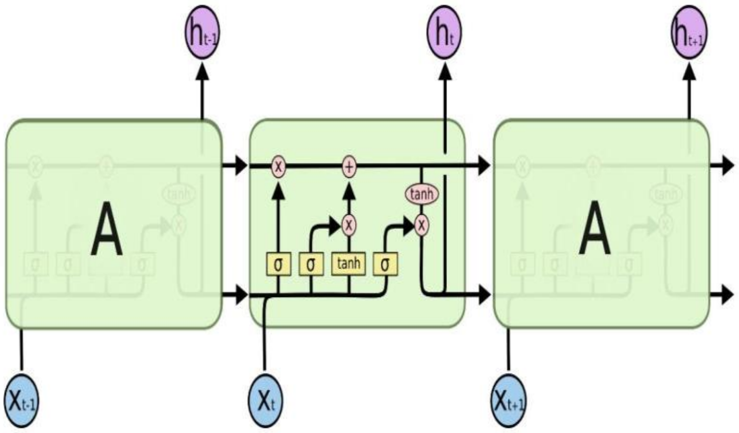
B. Recurrent Neural Nets (RNNs) Decoder Architecture:

Recurrent neural networks are artificial neural networks with a directed cycle of connections between the units. The benefit of using a RNN rather than a standard feed forward network is that it could handle any no of inputs using the memory. RNNs was discovered in 1980s by John Hopfield, the creator of the renowned Hopfield model. Recurrent neural networks, in easy term, are network with loop that allows information to persist in networks.

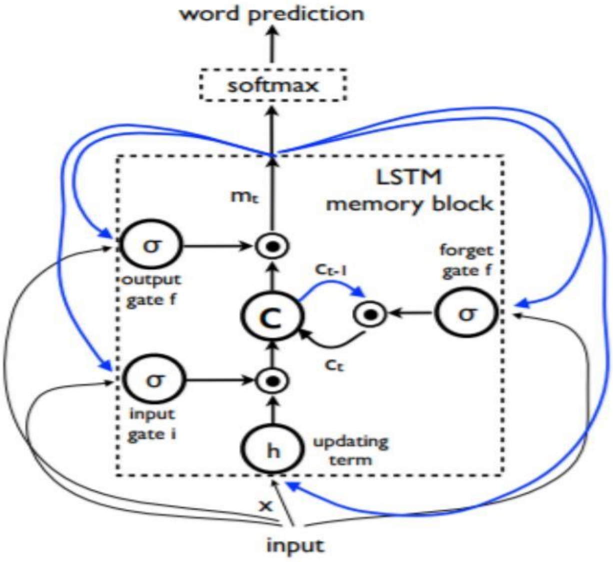


RNNs have a number of flaws, one of which is that theydon't account for long-term interdependence.

Consider a system that attempts to compose phrases on its own. For example, if machine is attempting to predict the last words in phrase English, it must first recognise the languages name that has been follow by fluent was contextually depended on term England. Traditional RNNs may fail if there is a huge gapbetween relevant data and when it is needed. Hochreier and Schmiduber propose Long Short Term Memory (LSTM) networks as a solution to this problem in 1997. this problem." Problem of the "long term dependencies." LSTM network have changed speech recognitions, machine translations,and other field since then. LSTMs, like traditional RNNs, have a chain;like structures, however that repeating module in an LSTM network have a distinctstructure. A simple LSTM network is demonstrated.



The horizontal lines runninng across the top, also known as cell state, is key to the LSTM network.
The cell state is maintained throughout all repeating modules and updated by gates at each module. The information in an LSTM network persists as a result of this.
With a small tweak, we employ this LSTM network. The LSTM network's architecture is shown in the diagram below.



The following equations regulate the entire network.

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}),$$

where it is input gate at the time t and W is training parameter. That variable mt1 denoted the module's output at time t-1 & reflects the sigmoid operations, which produces integers between 0 and 1, indicating how much of every components should be allowed through.

$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_{t-1}\right),$$

where ft is forget gate that controls whether the present cell value was forgotten.

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}),$$

where ot is output gate that selects whether or not to outputs the new cells value.

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}),$$

where ct is cell state which runs through all modules and ⊙ represents the tensor product in a gate values.

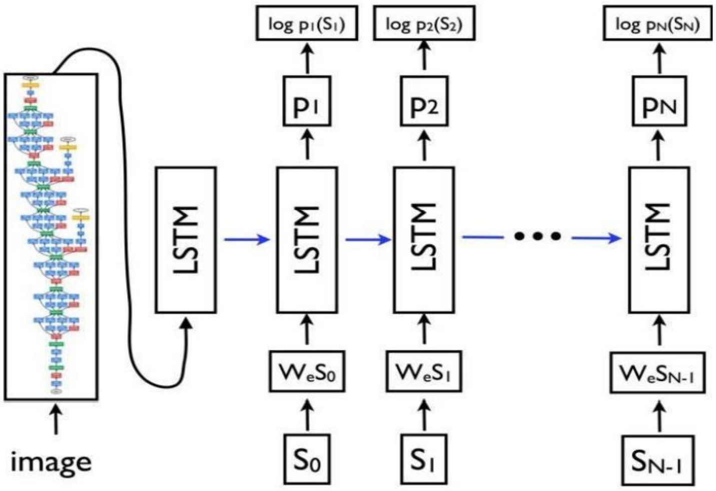
$$m_t = o_t \ \odot \ c_t,$$

where mt is the encoded vector which is then fed into the softmax function.

$$p_{t+1} = Softmax(m_t)$$

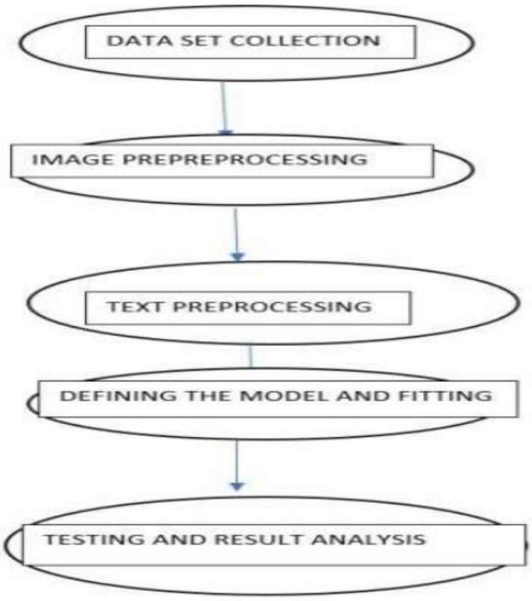
The word prediction is given by a module's output pt+1. The identical LSTM network was repeated till the network encounters a end token (.). The caption for a given image is generated by a succession of these word predictions. The entire training method for those combine model (CNN encoder + RNNs language generators) as well as the LSTM network is outlined below.

After seeing the image and all preceding words, the LSTM model was trained to forecast each word of the sentences as described by p(St|I, S0,..., St1).



C. IMPLEMENTATION DETAILS:

In this work, we will explain how to use the CNN-LSTM model for image captioning. Encoding is done with CNN Architecture, and decoding is done with LSTMs. We will now train the model with these two parameters as input once the image is given to CNN (Convolutional Neural Network), which extracts the image features with the help of vocabulary that is constructed using training captions data. We will put the model to the test after it has been trained. The flow diagram of our proposed model in this paper is shown below:



Data Collection

For the image caption generator, we'll be using the Flickr_8K dataset. There are also other big datasets like Flickr30K and MSCOCO datasets but it can takes week just to coach thenetwork so we will be using a small Flickr_8k dataset. The advantage of an enormous dataset is that we will build better models. Flickr_8k dataset contain a variety of image depicting scene and situations

The Flickr8k dataset contains 8k pictures with a total of 5 captions per image.

- 6k photos in training set
- 2k photos in test set

We also receive certain text file in relating to photograph in addition to image.

"Flickr8k.token.txt" was one of the file, and it contains the name of each images as well in their 5 captions. The information is presented in the "Flickr8k.token.txt" file as image name>-i captions>, i.e. the image name, captions number,& captions.

Data Cleaning:

To make the model more resilient to outliers and make fewer mistakes, the captions were cleaned as follows:

1. All words are transformed to lower case.
2. Eliminating punctuation marks.
3. Remove words with a length of less than two.
4. Special symbols such as @,,#, and others can be removed.

Only those terms that appear greater than or equal to the corpus's threshold. If we set the threshold to 10, we get 8424 words in the vocabulary and 373837 total words.

Loading of Training and Testing sets:

The reference to the photo in training the set can be found in the text files "Flickr 8k.trainImages.txt." All captions must include the following two tokens:

<s> —> Every caption will include a start sequence token at the beginning.

<e> —> At the end of each caption, an end sequence token will be inserted.

The photographs in the testing set are referenced in the text file "Flickr 8k.testImages.txt."

Pre-Processing Image Data:

A vector is used to supply input to a Machine Learning model. Each image must be transformed into the fixed-size vector that can be provided as input to corresponding neural network to make the model's processing and prediction process easier. Using the ResNet50 model, we can perform transfer learning. We get a fixed length relevant vector for each image using automated feature engineering. To obtain a vector of length 2048, remove the last softmax layer from the model.

We must preprocess the photographs after stacking the information sets in arrange to utilize them as input to the CNN. We must resize each picture to the same measure, 224X224X3, since we cannot bolster diverse measured photographs through the Convolution layer like CNN. We're too changing over the photographs to RGB utilizing the cv2 library's built-in capabilities.

Pre-Processing Text Data:

Word by word, the caption will be anticipated. Captions are used as target variables in the model's learning process. Each word was encode to a fixed size vector because the caption will be predicted one word at a time. We need to preprocess the captions for the photos after importing them using the FLICKR text data set so that there's no uncertainty or trouble when creating lexicon from the captions and preparing the profound learning show. We must to begin with decide whether the captions contain any numbers, and in the event that so, they must be killed, taken after by the evacuation of white spaces and missing captions within the given information set

A pretrained GLOVE word embedding model is used to map each word/index to a 50-length vector. Each sequences has 35 indice, in each of which is 50dimensional vectors.

As results, x = 45*50 = 1750.

The maximum length of a caption provided in the training data is 35 characters.

To avoid equivocalness during vocabulary building and demonstrate preparing, we have to be change all upper case letters within the captions to lower case. Since this show produces captions one word at a time, and already made words are utilized as inputs with picture traits, they are connected at the starting and conclusion of each caption to tell the neural arrange approximately the begin and conclusion of the caption.

DEFINING THE MODEL AND FITTING:

Functional API is used to integrate Models since the input consists of a partial caption and an image vector. After gathering the data and preparing the photos and descriptions, as well as developing vocabulary. We must now define the models for caption generation.

input_image (224x224 —> 2048 —> 256 dimensions) input_caption(batch_size x 35 —> batch_size x 35 x 50 —> LSTM —> 256 dimensions) input_image + input_caption -> 256 dimensions -> 1848 dimensions -> softmax -> probable_word

The Python SciPy environment was used to implement the model. Because of the presence of the VGG net, which is used for object identification, Keras 2.0 is employed to create that deep learning model. For developing and training deep neural network, the Tensorflow library was installed as backend for that Keras framework. Google developed TensorFlow, an deep learning library. It provides heterogeneous platform for algorithm execution, meaning it can run on low-powered device such as mobile phone as well as large:scale distributed system with thousands of GPU. The neural network was trained on the 640 Cuda core Nvidia Geforce 1050 graphics processor unit. TensorFlow employs graph definition to specify the structure of our network. Once a graph has been defined, it can be run on any device that supports it. The photo features are computed and saved using the pre-trained model. To eliminate the duplication of putting each photo through the network every time we wish to test a different language model configuration, these attributes are loaded into the model as that interpretation of certain photo in that dataset.The picture feature are also pre-loaded in order to deploy that image captioning model real time. Figure 1 depicts the model's architecture.

D. RESULTS AND COMPARISIONS:

The picture captioning model was implemented, & we were able to generate caption that was somewhat compared to those generate by human. The VGG net model gives probability to all the items which may be present in image. The images is converted into word vector by the model. This word vector was fed into LSTM cells, that generate a phrase from it.

The BLEU score is used to determine the model's correctness. The bilingual evaluation understudy (BLEU) algorithm assesses the quality of material that has been machine translated. It was one of the first metrics to show a strong link to human judgement.

The blue score is always between 0 and 1, with 0 indicating that the machine translation is unrelated to the reference sentence. Two inputs are required by the BLEU evaluation system:

(i) a numerical translation similarity metric was assigned and measured against.

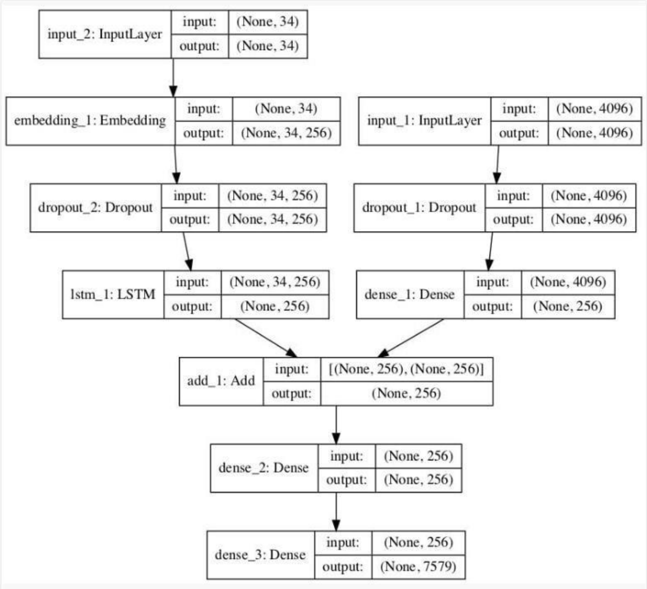
(ii) a corpus of human references translation.

To calculate the BLUE score, we construct caption for all of the test image first, and then use those caption as input sentences. We compare those candidate sentence to 5 human caption & average the BLEU score for the candidte for each reference. Using that Natural Language Toolkit (NLTK) Python library, we compute 1000 BLEU scores for 1000 test photos.

These BLEU scores were averaged across 1000 test photos. After 70 epochs of training with a batch size of 512, the model's net BLEU score was found to be 0.452, or 45.2 percent, while the state of that art on Flickr8k was about 68 percent. We could get close to state-of-the-art result by increasing that number of epochs, but it would take more computation. Reducing the batch size can also enhance the net BLEU score.

OUTPUTS:

In the majority of situations, the programme were able to predict the meaningful captions in an image. However, it became confused in some circumstances due to a shortage of tokens in the dictionary to characterise an event.



"two young girls are playing with lego toy."

CONCLUSION

This end-to-end system i.e neural network system was capable of viewing a picture & creating an feasible English descriptions based on words in the vocabulary derived from tokens in train image captions. A convolution neural network encoded and an LSTM decoded are included in the model, that aid in sentence production. This model's goal is to maximise the sentence's likelihood given that image.

Experiment with that model on the Flickr8K dataset yielded promising result. The BLEU score is used to determine the model's correctness. If the same model is used to a larger dataset, the accuracy can be improved. It will also be fascinating to examine how unsupervised data, both from photos & text, may be used in improving image description algorithms.

FUTURE PROSPECTS

The model accuracy could be improve by deploying it on a larger datasets, which increases the model's vocabulary dramatically. The adoption of relatively modern architecture, such as ResNet and GoogleNet, can help improve classification accuracy, lowering the language generation error rate.

Aside from that, using a bidirectional LSTM network and a Gated Recurrent Unit could assist improve the model accuracy.

REFERENCES

[1] Oriol Vinals ,Alexander Toshev, Samy Bengio,Dumitru Erhan,” Show and Tell: A Neural Image Caption Generator”,[2020]

[2] Kelvin Xu Jimy Lei Ba Ryan Kiros Kyunghyun Cho Aaron Courville Ruslan Salakhutdinov Richard S. Zemel Yoshua Bengio,”
Show, Attend and Tell: Neural Image Caption Generaton with Visual Attention”,[2019]

[3] Yupn Hang, Be Liu , Jialong Fu, Yuton Lu ,” A Picture is Worth a Thousand Word: A Unified System for Diverses
Captions and Rich Images Generation”,[2018]

[4] Paul Hongsuck Seo, Piyush Sharma Tomer Levinboim, Bohyung Han , Radu Soricut,” Reinforcing an Image Caption Generator
Using Off-Line Human Feedback” ,[2019]

[5] Shizhe Chen , Qin Jin , Peng Wang , Qi Wu,” Say As You Wish: Finegrained Control of Image Caption Generation with Abstract Scene Graphs” ,[2019]

[6] Hamed R. Tavakoli , Rakshith Shetty, Ali Borji, Jorma Laaksonen,” Paying Attention to Descriptions Generated by Image Captioning Models”,[2018]

[7] Alexander Mathews, Lexing Xie, Xuming He,” Learning to Generate Stylised Image Captions using Unaligned Text”,[2019]

[8] Xiaoqiang Lu, Binqiang Wang, Xiangtao Zheng, Xulong Li ,” Exploring Model and Data for Remte Sensing Image Caption Generation”,[2018]

[9] Fenlin Lu , Xuncheng Ren , Yunxin Liu , Hofeng Wang and Xu Sun,” Stepwise Image-Topic Merging Network for Generating Details and Comprehensive Image Caption”,[2019]

[10] Chen Chen, Shai Mu, Wanpen Xio, Zexion Ye, Liesi Wu, Qi Ju,” Improving Images Captioning with Conditional Generatives Adversarial Net” ,[2020]